

ИНСТИТУТ ИНТЕЛЛЕКТУАЛЬНЫХ КИБЕРНЕТИЧЕСКИХ СИСТЕМ

КАФЕДРА КИБЕРНЕТИКИ

ОДОБРЕНО

**РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ**

**ФУНКЦИОНАЛЬНОЕ ПРОГРАММИРОВАНИЕ (FUNCTIONAL PROGRAMMING)**

Направление подготовки  
(специальность)

[1] 09.03.04 Программная инженерия

Семестр	Трудоемкость, кред.	Общий объем курса, час.	Лекции, час.	Практич. занятия, час.	Лаборат. работы, час.	В форме практической подготовки/В СРС, час.	КСР, час.	Форма(ы) контроля, экз./зач./КР/КП
4	2	72	10	20	0	42	0	3
Итого	2	72	10	20	0	42	0	

## АННОТАЦИЯ

Дисциплина формирует у студентов понимание принципов функционального программирования

### 1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

Целями освоения учебной дисциплины являются:

- изучение теоретических аспектов функционального программирования на примере абстрактного функционального языка и языка функционального программирования Haskell;
- изучение основ языка Haskell и привитие навыков практической работы с этим языком.

### 2. МЕСТО УЧЕБНОЙ ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП ВО

Данная дисциплина является курсом по выбору в цикле дисциплин направления.

Для успешного обучения требуются сложившиеся навыки работы с компьютером и успешное освоение следующие дисциплин:

- информатика;
- дискретная математика (математическая логика);
- дискретная математика (логические исчисления);
- дискретная математика.

Желательно знакомство с основами лямбда-исчисления.

### 3. ФОРМИРУЕМЫЕ КОМПЕТЕНЦИИ И ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

Универсальные и(или) общепрофессиональные компетенции:

Код и наименование компетенции	Код и наименование индикатора достижения компетенции
--------------------------------	------------------------------------------------------

Профессиональные компетенции в соответствии с задачами и объектами (областями знаний) профессиональной деятельности:

Задача профессиональной деятельности (ЗПД)	Объект или область знания	Код и наименование профессиональной компетенции; Основание (профессиональный стандарт-ПС, анализ опыта)	Код и наименование индикатора достижения профессиональной компетенции
производственно-технологический			
- освоение и применение средств автоматизированного проектирования, разработки, тестирования и сопровождения	- программный продукт (создаваемое программное обеспечение) - процессы жизненного	ПК-3 [1] - способен применять навыки использования различных технологий разработки программного обеспечения	З-ПК-3[1] - Знать различные технологии разработки программного обеспечения; У-ПК-3[1] - Уметь применять различные

<p>программного обеспечения; - освоение и применение методов и инструментальных средств управления инженерной деятельностью и процессами жизненного цикла программного обеспечения; - использование типовых методов для контроля, оценки и обеспечения качества программной продукции; - обеспечение соответствия разрабатываемого программного обеспечения и технической документации российским и международным стандартам, техническим условиям, ведомственным нормативным документам и стандартам предприятия; - участие в процессах разработки программного обеспечения</p>	<p>цикла программного продукта - методы и инструменты разработки программного продукта</p>	<p><i>Основание:</i> Профессиональный стандарт: 06.001, 06.011</p>	<p>технологии разработки программного обеспечения; В-ПК-3[1] - Владеть навыками использования различных технологий разработки программного обеспечения</p>
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------	------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------

#### 4. ВОСПИТАТЕЛЬНЫЙ ПОТЕНЦИАЛ ДИСЦИПЛИНЫ

Направления/цели воспитания	Задачи воспитания (код)	Воспитательный потенциал дисциплин
Профессиональное воспитание	Создание условий, обеспечивающих, формирование ответственности за профессиональный выбор, профессиональное развитие и профессиональные решения (В18)	Использование воспитательного потенциала дисциплин профессионального модуля для формирования у студентов ответственности за свое профессиональное развитие посредством выбора студентами индивидуальных образовательных траекторий, организации системы общения между всеми участниками образовательного процесса, в том числе с использованием новых

<p>Профессиональное воспитание</p>	<p>Создание условий, обеспечивающих, формирование научного мировоззрения, культуры поиска нестандартных научно-технических/практических решений, критического отношения к исследованиям лженаучного толка (В19)</p>	<p>информационных технологий.</p> <p>1.Использование воспитательного потенциала дисциплин/практик «Научно-исследовательская работа», «Проектная практика», «Научный семинар» для:</p> <ul style="list-style-type: none"> <li>- формирования понимания основных принципов и способов научного познания мира, развития исследовательских качеств студентов посредством их вовлечения в исследовательские проекты по областям научных исследований.</li> </ul> <p>2.Использование воспитательного потенциала дисциплин "История науки и инженерии", "Критическое мышление и основы научной коммуникации", "Введение в специальность", "Научно-исследовательская работа", "Научный семинар" для:</p> <ul style="list-style-type: none"> <li>- формирования способности отделять настоящие научные исследования от лженаучных посредством проведения со студентами занятий и регулярных бесед;</li> <li>- формирования критического мышления, умения рассматривать различные исследования с экспертной позиции посредством обсуждения со студентами современных исследований, исторических предпосылок появления тех или иных открытий и теорий.</li> </ul>
<p>Профессиональное воспитание</p>	<p>Создание условий, обеспечивающих, формирование профессионально значимых установок: не производить, не копировать и не использовать программные и технические средства, не приобретённые на законных основаниях; не нарушать признанные нормы авторского права; не нарушать тайны передачи сообщений, не практиковать вскрытие информационных систем и сетей передачи данных;</p>	<p>1. Использование воспитательного потенциала дисциплин "Информатика (Основы программирования)", Программирование (Объектно-ориентированное программирование)", "Программирование (Алгоритмы и структуры данных)" для формирования культуры написания и оформления программ, а также привития навыков командной работы за счет использования систем управления проектами и контроля версий.</p>

	<p>соблюдать конфиденциальность доверенной информации (B40)</p>	<p>2.Использование воспитательного потенциала дисциплины "Проектная практика" для формирования культуры решения изобретательских задач, развития логического мышления, путем погружения студентов в научную и инновационную деятельность института и вовлечения в проектную работу.</p> <p>3.Использование воспитательного потенциала профильных дисциплин для формирования навыков цифровой гигиены, а также системности и гибкости мышления, посредством изучения методологических и технологических основ обеспечения информационной безопасности и кибербезопасности при выполнении и защите результатов учебных заданий и лабораторных работ по криптографическим методам защиты информации в компьютерных системах и сетях.</p> <p>4.Использование воспитательного потенциала дисциплин "Информатика (Основы программирования)", Программирование (Объектно-ориентированное программирование)", "Программирование (Алгоритмы и структуры данных)" для формирования культуры безопасного программирования посредством тематического акцентирования в содержании дисциплин и учебных заданий.</p> <p>5.Использование воспитательного потенциала дисциплины "Проектная практика" для формирования системного подхода по обеспечению информационной безопасности и кибербезопасности в различных сферах деятельности посредством исследования и перенятия опыта постановки и решения научно-практических задач организациями-партнерами.</p>
--	-----------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## 5. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

Разделы учебной дисциплины, их объем, сроки изучения и формы контроля:

№ п.п	Наименование раздела учебной дисциплины	Недели	Лекции/ Практ. (семинары) / Лабораторные работы, час.	Обязат. текущий контроль (форма*, неделя)	Максимальный балл за раздел**	Аттестация раздела (форма*, неделя)	Индикаторы освоения компетенции
	<i>4 Семестр</i>						
1	Раздел 1	1-5	5/10/0		36	ДЗ-5	З-ПК-3, У-ПК-3, В-ПК-3
2	Раздел 2	6-10	5/10/0		24	ДЗ-10	З-ПК-3, У-ПК-3, В-ПК-3
	<i>Итого за 4 Семестр</i>		10/20/0		60		
	<b>Контрольные мероприятия за 4 Семестр</b>				40	3	З-ПК-3, У-ПК-3, В-ПК-3

\* – сокращенное наименование формы контроля

\*\* – сумма максимальных баллов должна быть равна 100 за семестр, включая зачет и (или) экзамен

Сокращение наименований форм текущего контроля и аттестации разделов:

Обозначение	Полное наименование
ДЗ	Домашнее задание
З	Зачет

## КАЛЕНДАРНЫЙ ПЛАН

Недел	Темы занятий / Содержание	Лек.,	Пр./сем.	Лаб.,
-------	---------------------------	-------	----------	-------

и		час.	, час.	час.
	<i>4 Семестр</i>	10	20	0
<b>1-5</b>	<b>Раздел 1</b>	5	10	0
1	<b>Темы 1. Введение в функциональное программирование. Теоретические основы.</b> Парадигма программирования. Императивное и декларативное программирование. Функциональное программирование, композиция функций. Формальная система. Элементы лямбда-исчисления (объект в лямбда-исчислении, расстановка скобок, понятие связанной и свободной переменной, применение функции, понятия связанной и свободной переменной, подстановка, конверсия объектов, редукция объектов). Понятие комбинатора, комбинаторная характеристика. Рекурсия в лямбда-исчислении, комбинатор Y. Простое типизированное лямбда-исчисление. Типизация по Чёрчу и по Карри.	Всего аудиторных часов		
		1	2	0
		Онлайн		
		0	0	0
2	<b>Тема 2-3. Основы языка Haskell. Рекурсия. Списки.</b> Свойств языка Haskell (функциональность, лаконичность, ленивость, строгая типизация, модульность). REPL. команды интерпретатора ghci. Константы, переменные, операторы. Синтаксис типов. Ограничения типов. Синтаксис определений, аннотации типов. Основные типы данных: числа, логические значения, списки (синтаксис, генераторы списков, ленивость), кортежи (отличие от списков). Функции: сопоставление с образцом, охранные выражения. Виды рекурсии (функциональная, рекурсивные типы данных). Функциональная рекурсия: рекурсия вместо циклов. Примеры рекурсивных функций. Хвостовая рекурсия. Вычисления с аккумулятором. Списки как рекурсивный тип данных. Функции над списками.	Всего аудиторных часов		
		1	2	0
		Онлайн		
		0	0	0
3	<b>Тема 4-5. Каррированные функции. Функции высшего порядка. Производительность рекурсивных алгоритмов. Мемоизация.</b> Способы представления функции от нескольких переменных. Понятие о каррированной функции. Функции высшего порядка. Оператор каррирования. Функции высшего порядка над списками. Лямбда-выражения. Операторы (\$) и (.) для упрощения записи выражений. Производительность функции fib. Принципе разделяй и властвуй, оценка сложности. Динамическое программирование. Мемоизация.	Всего аудиторных часов		
		1	2	0
		Онлайн		
		0	0	0
4	<b>Тема 6-7. Пользовательские типы данных в языке Haskell. Свёртки.</b> Синонимы типов. Алгебраические типы данных. Определение пользовательских алгебраических типов данных: конструктор значения типа данных, именованные типы, конструктор типа данных, рекурсивные типы	Всего аудиторных часов		
		1	2	0
		Онлайн		
		0	0	0

	данных. Пример рекурсивных типов данных: бинарные деревья. Определение типа бинарных деревьев, функции добавления элемента, сортировка деревом. Правая свёртка списка. Функции с аккумулятором и левая свёртка списка. Равенство левой и правой свёрток. Префиксные суммы. Свёртка рекурсивного значения рекурсивного типа данных.			
5	<b>Тема 8-9. Моноиды. Классы типов. Примеры классов типов.</b> Формальная алгебра. Моноид. Гомоморфизм моноида. Списочный гомоморфизм. Понятие класса типов. Пример класс типа Eq. Определение класса типа. Экземпляр класса типа. Класс типов Monoid. Использование newtype для определения нескольких экземпляров классов типов.	Всего аудиторных часов		
		1	2	0
		Онлайн		
		0	0	0
6-10	<b>Раздел 2</b>	5	10	0
6	<b>Тема 10-11. Категории. Функтор. Аппликативный функтор.</b> Сведения из теории категорий: объект, морфизм, композиция, тождественный морфизм, функтор. Категория типов Haskell. Эндофунктор в категории типов Haskell. Класс типов для Functor для функторов. Законы функторов. Класс типов Applicative для аппликативных функторов. Законы аппликативного функтора. Использование функтора вместе с аппликативным функтором. ZipList.	Всего аудиторных часов		
		1	2	0
		Онлайн		
		0	0	0
7	<b>Тема 12-13. Монады. Ввод-вывод.</b> Класс типов Monad для монад. Связь с функтором и аппликативным функтором. Стандартные монады: Maybe, List, Reader, Writer, State, IO. do – нотация.	Всего аудиторных часов		
		1	2	0
		Онлайн		
		0	0	0
8	<b>Тема 14-15. Applicative и MonadPlus. Обработка ошибок. Трансформеры монад. Расширения Haskell.</b> Классы типов Alternative и MonadPlus. Использование в do-нотации. Тип Either. Обработка ошибок с типом Either. Монада Except. Паргама. Фундепсы. Комбинирование эффектов монад: трансформеры монад. Стандартные трансформеры монад.	Всего аудиторных часов		
		1	2	0
		Онлайн		
		0	0	0
9 - 10	<b>Тема 16. Доступ к части сложного значения. Линзы.</b> Понятие линзы к двух функций view и set. Недостаток введённого определения. Функция over. Выражение set через over. Линзы с эффектами. Линзы Ван Лааховена. Реализация функций view, set и over.	Всего аудиторных часов		
		2	4	0
		Онлайн		
		0	0	0

Сокращенные наименования онлайн опций:

Обозначение	Полное наименование
ЭК	Электронный курс
ПМ	Полнотекстовый материал
ПЛ	Полнотекстовые лекции
ВМ	Видео-материалы
АМ	Аудио-материалы
Прз	Презентации

Т	Тесты
ЭСМ	Электронные справочные материалы
ИС	Интерактивный сайт

## ТЕМЫ ПРАКТИЧЕСКИХ ЗАНЯТИЙ

Недели	Темы занятий / Содержание
	<i>4 Семестр</i>
1 - 2	<b>Элементы лямбда-исчисления.</b> Задачи на определение связанных и свободных переменных, подстановку, редукцию объектов лямбда-исчисления. Задачи на типизацию объектов лямбда-исчисления: комбинаторы I, K, S, B, C.
3 - 4	<b>Введение в программирование на языке Haskell.</b> Примеры работы с интерпретатором ghci: означивание выражений в интерпретаторе, создание переменных, вывод типа выражений, загрузка исходного кода в интерпретатор. Программирование функций.
5 - 6	<b>Программирование на языке Haskell. Списки и рекурсия. Каррирование функций.</b> Задачи на использование списков, программирование рекурсивных функций: fib, factorial, reverse и т.д. Реализация операторов каррирования и декаррирования для функций 2, 3 и 4-х аргументов. Решения задач о балансировке скобок и списка треугольных чисел.
7 - 8	<b>Мемоизация. Пользовательские типы данных.</b> Реализация функции fib при помощи мемоизации. Реализация пользовательского типа данных: пар, комплексный чисел, бинарного дерева.
9 - 10	<b>Программирование с использованием свёрток. Моноиды.</b> Реализация функций высшего порядка над списками с использованием свёрток. Реализация аддитивного и мультипликативного моноида для чисел, для логических значений.
11 - 12	<b>Функтор. Аппликативный функтор.</b> Реализация функтор и аппликативного функтора для конструкторов типов данных: Maybe, List. Реализация ZipList. Решение задач без использования сравнения с образцом для типов данных, которые поддерживают интерфейс функтора и аппликативного функтора.
13 - 14	<b>Монады.</b> Решения задач, требующих чтения, изменения, либо полной поддержки состояний с использованием Monad.
15 - 16	<b>Ввод-вывод.</b> Решение задач ввода-вывода с использованием do-нотации и монад.

## 6. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

Основными используемыми образовательными технологиями являются лекции и практические занятия, на которых студенты прослушивают теоретический материал, решают предлагаемые задачи, консультируются с преподавателем относительно семестровых заданий.

Дополнительно на усмотрение преподавателя могут применяться презентационные технологии, предусматривающие использование компьютерной техники как преподавателем, так и студентами. В частности лекции могут сопровождаться демонстрациями слайдов и трансляцией работы преподавателя с интерпретатором функционального языка, а демонстрация готовых семестровых работ может выполняться на ноутбуках студентов.

## 7. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

Фонд оценочных средств по дисциплине обеспечивает проверку освоения планируемых результатов обучения (компетенций и их индикаторов) посредством мероприятий текущего, рубежного и промежуточного контроля по дисциплине.

Связь между формируемыми компетенциями и формами контроля их освоения представлена в следующей таблице:

Компетенция	Индикаторы освоения	Аттестационное мероприятие (КП 1)
ПК-3	З-ПК-3	З, ДЗ-5, ДЗ-10
	У-ПК-3	З, ДЗ-5, ДЗ-10
	В-ПК-3	З, ДЗ-5, ДЗ-10

## Шкалы оценки образовательных достижений

Шкала каждого контрольного мероприятия лежит в пределах от 0 до установленного максимального балла включительно. Итоговая аттестация по дисциплине оценивается по 100-балльной шкале и представляет собой сумму баллов, заработанных студентом при выполнении заданий в рамках текущего и промежуточного контроля.

Итоговая оценка выставляется в соответствии со следующей шкалой:

Сумма баллов	Оценка по 4-ех балльной шкале	Оценка ECTS	Требования к уровню освоению учебной дисциплины
90-100	5 – «отлично»	A	Оценка «отлично» выставляется студенту, если он глубоко и прочно усвоил программный материал, исчерпывающе, последовательно, четко и логически стройно его излагает, умеет тесно увязывать теорию с практикой, использует в ответе материал монографической литературы.
85-89	4 – «хорошо»	B	Оценка «хорошо» выставляется студенту, если он твёрдо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос.
75-84		C	
70-74		D	

65-69			Оценка «удовлетворительно» выставляется студенту, если он имеет знания только основного материала, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки, нарушения логической последовательности в изложении программного материала.
60-64	3 – «удовлетворительно»	E	
Ниже 60	2 – «неудовлетворительно»	F	Оценка «неудовлетворительно» выставляется студенту, который не знает значительной части программного материала, допускает существенные ошибки. Как правило, оценка «неудовлетворительно» ставится студентам, которые не могут продолжить обучение без дополнительных занятий по соответствующей дисциплине.

## 8. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

### ОСНОВНАЯ ЛИТЕРАТУРА:

1. ЭИ К88 Функциональное программирование : конспект лекций, Москва: НИЯУ МИФИ, 2020
2. ЭИ Ч-88 Функциональное программирование на C++ : учебное пособие, Москва: ДМК Пресс, 2020
3. 004 С32 Функциональное и логическое программирование : учебное пособие для вузов, Г. М. Сергиевский, Н. Г. Волченков , Москва: Академия, 2010

### ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА:

1. 004 Е42 Функциональное и логическое программирование : учебное пособие, Москва: Бинوم. Лаборатория знаний, 2011

### ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ:

Специальное программное обеспечение не требуется

### LMS И ИНТЕРНЕТ-РЕСУРСЫ:

1. Haskell (<http://Haskell.org>)

<https://online.mephi.ru/>

<http://library.mephi.ru/>

## 9. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

Специальное материально-техническое обеспечение не требуется

## 10. УЧЕБНО-МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ДЛЯ СТУДЕНТОВ

В качестве оценочного средства используется 100 бальная семестровая система, учитывающая посещаемость занятий, активность (выполнение домашних занятий). Каждый раздел проходит аттестацию.

Итоговый балл за каждый раздел составляется из суммы оценок за сданные домашние задания.

Максимальная оценка за домашнее задание составляет 12 баллов. Критерии его оценки приведены ниже. Оценка домашнего задания проводится индивидуально для каждого студента и зависит от полноты выполнения домашнего задания, а также от сдачи домашнего задания в срок.

В первом разделе курса содержится 3 домашних задания, второй раздел курса включает в себя 2 домашних задания.

Максимальный балл за первый раздел 36 баллов, максимальный балл за второй раздел 24 балла.

Баллы за раздел могут быть повышены за посещаемость и активность, но итоговый балл не может превышать максимальный балл за раздел.

Зачёт состоит в решении 2 практических задач на программирование на языке Haskell.

Методические указания по выполнению домашнего задания

За весь курс студент должен выполнить пять домашних работ по соответствующим темам курса. Задания распространяются на семинарских занятиях и в электронном виде по почте. Актуальное задание и методические указания, если такие требуются, содержатся в тексте самого домашнего задания.

Первые два домашних задания выполняются письменно и требуют предоставления отчёта.

Оформление отчета.

Отчеты оформляются в электронном виде в одном из форматов: Open Document, Microsoft Word, RTF, PDF, PostScript. Отчет должен содержать в себе:

- ФИО студента, номер группы;
- Номер и название домашней работы;
- Решения заданий из домашней работы.

Оставшиеся два домашних задания выполняются в виде файла с кодом на языке Haskell. Шаблон файла с кодом предоставляется вместе с домашним заданием.

Оформления файла с кодом:

- Файла должен именоваться по шаблону: Фамилия\_И\_О\_ГРУППА.hs

Файл с кодом должен соответствовать шаблону, приложенному к домашнему заданию. Допускается введение дополнительных определений, но не допускается удаление существующих определений или изменение аннотаций типа, а также подключение дополнительных модулей (если обратное не оговорено заданием), кроме тех, что уже подключены в шаблоне файла с кодом.

Оценка работы выставляется исходя из:

- Полноты решения домашнего задания;
- Правильности решения отдельных заданий из домашнего задания.
- Прохождения предоставленным кодом тестов для домашних заданий на языке Haskell.

До 8-й недели студентам даётся билет из 6 задач, общим весом 25 баллов. На 8-й неделе подводится итоговый контроль по результатам сдачи задач.

Пример билета для решения:

На зачёте студенту даётся для решения две практические задачи на языке Haskell.

Примеры возможных задач:

1. Создание конечных списков из  $N$  элементов:

Сконструировать список натуральных чисел.  $N = 20$

Допускается использование генераторов списков, определений типов, конструирующих функций. В случае использования функций привести пример вызова построенной функции.

2. Создание бесконечных списков:

Сконструировать бесконечный список факториалов.

Допускается использование генераторов списков, определений типов, конструирующих функций. В случае использования функций привести пример вызова построенной функции.

3. Построить функцию, вычисляющую  $N$ -ый элемент ряда:

$$F(x, n) = x^n$$

Дать пример вызова построенной функции и ожидаемого результата.

4. Функции работы со списками. Построить функцию:

`GetN(L, n)` — функция вычленения  $N$ -ого элемента из заданного списка

Дать пример вызова построенной функции и ожидаемого результата.

5. Функции с аккумулятором. Написать функцию из задания 3, но с использованием накапливающего параметра.

6. Функции с аккумулятором. Написать функцию из задания 4, но с использованием накапливающего параметра.

7. Сконструировать конечный список из  $N = 200$  натуральных чисел, используя конструктор списка.

Возможные ответы:

[1..200]

или

[1,2..200]

8. Реализуйте функцию, вычисляющую N-ый член ряда Фибоначчи, в виде хвостовой рекурсии.

9. Реализуйте экземпляр класса типов Monad для типа данных Either.

10. Реализуйте экземпляр класса типов Functor для типа данных Const.

11. Реализуйте экземпляр класса типов Functor для типа данных Identity.

## 11. УЧЕБНО-МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ДЛЯ ПРЕПОДАВАТЕЛЕЙ

В качестве оценочного средства используется 100 бальная семестровая система, учитывающая посещаемость занятий, активность (выполнение домашних занятий,). Каждый раздел проходит аттестацию.

Итоговый балл за каждый раздел составляется из суммы оценок за сданные домашние задания.

Максимальная оценка за домашнее задание составляет 12 баллов. Критерии его оценки приведены ниже. Оценка домашнего задания проводится индивидуально для каждого студента и зависит от полноты выполнения домашнего задания, а также от сдачи домашнего задания в срок.

В первом разделе курса содержится 3 домашних задания, второй раздел курса включает в себя 2 домашних задания.

Максимальный балл за первый раздел 36 баллов, максимальный балл за второй раздел 24 балла.

Баллы за раздел могут быть повышены за посещаемость и активность, но итоговый балл не может превышать максимальный балл за раздел.

Зачёт состоит в решении 2 практических задач на программирование на языке Haskell.

## Методические указания по выполнению домашнего задания

За весь курс студент должен выполнить пять домашних работ по соответствующим темам курса. Задания распространяются на семинарских занятиях и в электронном виде по почте. Актуальное задание и методические указания, если такие требуются, содержатся в тексте самого домашнего задания.

Первые два домашних задания выполняются письменно и требуют предоставления отчёта.

Оставшиеся два домашних задания выполняются в виде файла с кодом на языке Haskell. Шаблон файла с кодом предоставляется вместе с домашним заданием.

Автор(ы):

Маренков Александр Владимирович

Рословцев Владимир Владимирович